CUMENTATION PAGE

2

2. REPORT DATE

FINAL/30 SEP 89 TO 29 SEP 92

4. TITLE AND SUBTITLE

ADAPTIVE NETWORKS FOR SEQUENTIAL
DECISION PROBLEMS (U)

6. AUTHOR(S)

Professor Andrew Barto

2305/B3
AFOSR-89-0526

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Massachusetts
Dept of Computer Sciences
Amherst MA 01003

AFOSR-TR-

9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR/NM
110 DUNCAN AVE, SUTE B115
BOLLING AFB DC 20332-0001

10. SPONSORING MONITORING AGENCY REPORT NUMBER

AFOSR-89-0526

11. SUPPLEMENTARY NOTES

14P

12a. DISTRIBUTION AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

UL

13. ABSTRACT (Maximum 200 words)

Considerable progress was made in developing artificial neural network methods for solving stochastic sequential decision problems. The research focused on reinforcement learning methods based on approximating dynamic programming (DP). They used problems in the domains of robot fine motion control, navigation, and steering control in order to develop and test learning algorithms and architectures. Although most of these problems were simulated, they also began to apply DP-based learning algorithms to actual robot control problems with considerable success. Progress was made on reinforcement learning methods using continuous actions, modular network architectures, and architectures using abstract actions. Theoretical progress was made in relating DP-based reinforcement learning algorithms to more conventional methods for solving stochastic sequential decision problems. As a result of this research there is an improved understanding of these algorithms and how they can be successfully used in applications.

14. SUBJECT TERMS

15. NUMBER OF PAGES

13

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR(SAME AS REPORT)

FINAL TECHNICAL REPORT: AFOSR-89-0526

"Adaptive Networks for Sequential Decision Problems"

Principal Investigator: Andrew G. Barto

Department of Computer Science
University of Massachusetts, Amherst

Summary—Considerable progress was made in developing artificial neural network methods for solving stochastic sequential decision problems. Our research focused on reinforcement learning methods based on approximating dynamic programming (DP). We used problems in the domains of robot fine motion control, navigation, and steering control in order to develop and test learning algorithms and architectures. Although most of these problems were simulated, we also began to apply DP-based learning algorithms to actual robot control problems with considerable success. Progress was made on reinforcement learning methods using continuous actions, modular network architectures, and architectures using abstract actions. Theoretical progress was made in relating DP-based reinforcement learning algorithms to more conventional methods for solving stochastic sequential decision problems. As a result of this research, we have a much improved understanding of these algorithms and how they can be successfully used in applications.

1 Introduction

Following is the summary of the research proposal that led to funding of the research being reported here. It states the research objectives.

> This project seeks to develop learning methods for artificial neural networks (or connectionist networks) for application to problems formalized as *stochastic sequential decision problems*. In these problems the consequences of network actions unfold over an extended time period after an action is taken, so that actions must be selected on the basis of both their short-term and long-term consequences and under uncertainty. Problems of this kind can be viewed as discrete-time stochastic control problems. The theory of stochastic sequential decision making and the computational techniques associated with it, known as *stochastic dynamic programming*, provide ways of understanding the capabilities of the reinforcement-learning and temporal credit-assignment methods we previously developed and suggest a variety of extensions to them which can be implemented as adaptive networks. These extensions involve *model-based* and *hierarchical* learning. The long-term goal of this research is the development

of network methods for the efficient solution of stochastic sequential decision problems in the absence of complete knowledge of underlying dynamics.

We made considerable progress in furthering the development of DP-based reinforcement learning algorithms and in understanding their properties and domains of utility. Below we describe our major accomplishments. Some aspects of this project were closely related to research funded under National Science Foundation Grant ECS-8912623.

2 Reinforcement Learning of Continuous Values

Part of our research addressed methods for allowing networks with continuous outputs to learn via reinforcement learning. Although this work did not explicitly rely on the formalism of sequential decision problems, it addressed a capability that learning systems must have for a wide range of such problems. Whereas most reinforcement learning systems are restricted to a finite set of actions, many sequential decision problems require learning over a continuous range of actions. Our effort focused on Stochastic Real-Valued (SRV) units, which are neuron-like units with real-valued outputs that can be trained via reinforcement feedback. SRV units were developed by V. Gullapalli with support from this grant and formed the basis of his Ph.D. dissertation (he received the Ph.D. in 1992). We conducted a number of experiments using SRV units in a simulated pole-balancing task and control of a simulated three degree-of-freedom robot arm in an underconstrained positioning task. Results indicated that networks using SRV units can learn these tasks faster than networks based on supervised learning. Gullapalli has published a journal article, several conference papers, and a book chapter on this work.

Gullapalli also used SRV units in a neural network model of perception by training a network with SRV units to model area 7a of the posterior parietal cortex, a cortical area thought to transform visual stimuli from retinotopic coordinates into a head-centered coordinate system [5]. Results showed that the SRV network reproduces the performance of previous models while being free of some of their limitations with respect to biological plausibility.

Based on the promise shown by these simulations, we applied a network using SRV units to the problem of robot peg in-hole insertion using a robot arm (a Zebra Zero). We achieved very promising results, described in refs. [7; 6]. This task is important in industrial robotics and is widely used by roboticists for testing approaches to robot control. Real-world conditions of uncertainty and noise can substantially degrade the performance of traditional control methods. Sources of uncertainty and noise include (1) errors and noise in sensations, (2) errors in execution of motion commands, and (3) uncertainty due to movement of the part grasped by the robot. Under such conditions, traditional methods do not perform very well, and the peg-insertion problem becomes a good candidate for adaptive methods. For example, in the robot we used there is a large discrepancy between the sensed and actual positions of the peg under an external load similar to what can occur during peg insertion: whereas the actual change in the peg's position under the external load was on the order of

2 to $3mm$, the largest sensed change in position was less than $0.025mm$. In comparison, the clearance between the peg and the hole was $0.175mm$.

Although it is difficult to design a controller that can robustly perform peg insertions despite the large uncertainty in sensory input, our results indicate that direct reinforcement learning can be used to learn a reactive control strategy that works robustly even in the presence of such a high degree of uncertainty. In a 2D version of the task (basically, inserting a peg into a narrow slot) the controller was consistently able to perform successful insertions within 100 time steps after about 150 learning trials. Furthermore, performance as measured by insertion time continued to improve, decreasing continuously over learning trials. The controller became progressively more *skillful* at peg insertion with training. Similar results were obtained in a 3D task although learning took somewhat more trials.

Our experiences with this problem helped develop the following perspective on an important issue in control. The issue is when to approach a difficult control problem by first attempting to construct an accurate model of the system being controlled, versus when to attempt to solve the problem directly, i.e., without such a model. We argue that for *some* problems constructing an adequate model is actually more difficult than solving the problem itself. In robotics, it is a model of the task, e.g., a manipulation task, that is often problematic, not a model of the robot itself. Adaptive control methods appealing directly to the demands of the real task instead of to a model of the task can be very effective in such problems.

3 Navigation and Steering Control

Navigation and steering control problems provide useful test beds for exploring reinforcement learning algorithms for sequential decision problems. The basic form of these problems is that some kind of "vehicle" must move to a goal region of its environment while avoiding obstacles. Learning is used to improve the vehicle's performance with successive trials in terms of the distance traveled, the time required to reach the goal region, or other criteria. We have restricted attention to problems in which the environment is static in that it does not contain moving obstacles or other vehicles. By learning to navigate we mean learning the direction the vehicle should move from each location in order to reach the goal region along successively better paths. By learning to "steer," on the other hand, we mean learning to control a dynamic vehicle (for example, a vehicle that has mass and inertia), so that it reaches the goal region via successively more efficient trajectories. Often we are only interested in reaching the goal region in the minimum amount of time. Navigation and steering control also apply to more abstract spaces, such as the configuration space of a robot manipulator, instead of two- or three-dimensional cartesian space. Many different versions of these problems exist depending on the sensory and motor capabilities of the vehicle and on the structure of the underlying space.

Although navigation and steering control have obvious practical applications, we have used abstract versions of these problems as tools for helping us understand and refine DP-based reinforcement learning algorithms. However, our work is relevant to realistic examples of these problems, and some of our recent research, as well as research in other groups, experiments with these methods in actual navigation and steering control problems

3.1 Navigation

We developed a navigation test-bed simulating the movement of a cylindrical robot with a sonar belt in a planar environment. This test-bed was first used to study short-range homing in the presence of obstacles, that is, going to a "home" place from an arbitrary starting place within a neighborhood of the home place. The simulated robot has 16 distance sensors and 16 grey-scale sensors evenly placed around its perimeter. Thus, the input to the learning system at any time is a "sensation" vector of 32 real numbers representing its current view of the environment. (Other versions of this test-bed used fewer simulated sensors). This contrasts with various "grid-world" navigation problems that we have studied in the past, and that other groups are studying, in which the robot moves from square to square in a discretized environment.

This test-bed was used to illustrate the behavior of several DP-based learning architectures. One architecture was developed by J. Bachrach [1; 2]. It takes a structured approach to the problem and utilizes a priori knowledge of how local changes in position tend to change the robot's view. The homing aspect of the task and the obstacle avoidance aspect are handled by separate modules, implemented as "adaptive critics" that improve "evaluation landscapes" with experience. An evaluation landscape in this case is a real-valued function of the space of possible sensations; the higher the value of a sensation, the more the robot desires to be there. One critic learns to produce a gradually sloping evaluation landscape with a maximum at the home place. The other critic learns to place evaluation minima around obstacles. Gradient descent in the evaluation landscape formed by the superposition of the landscapes implemented by the two critics produces a trajectory that both avoids obstacles and moves towards home. This is related to the technique of potential functions, but differs in that it is perceptually-based and involves learning. That is, the evaluation landscape, which is improved through experience, only evaluates sensations directly; it does not directly evaluate places in space. Places indirectly receive evaluation according to the sensations that the robot would receive if it moved to them. Thus, the robot does not have to maintain a "bird's eye" view of the environment. This navigation control architecture is described in Bachrach's Ph.D. dissertation, completed in 1991. This work was our first experience with using reinforcement learning in a control scheme that is "behavior-based" in the sense of coordinating several different behaviors (homing and obstacle avoidance).

This test-bed was also used to illustrate a modular learning architecture developed by S. Singh [11] that leains several different homing/obstacle avoidance tasks in the same environment. This is discussed below in the section on modular architectures

3.2 Steering Control

To study steering control, we adpoted the "race track problem" where a starting line and a finish line are given in a two-dimensional workspace, along with two curves connecting corresponding edges of the starting and finish lines. The two curves represent the two side walls of the race track, and the region enclosed by the walls and the starting and finish lines is the admissible region of the workspace. As a "vehicle" we basically use a unit mass with no damping and stiffness. The controller applies bounded forces at discrete time intervals on the mass. The objective is to push it from the starting line to the finish line in minimum time without hitting the walls. Hitting a wall at any point is considered as controller failure. There are no constraints on the velocity at the finish line, so that any crossing of the finish line is regarded as success. The difficulty of this problem can be adjusted by the selection of the race track size and shape, the bound on controller forces, and the mass of the vehicle. The problem can be made stochastic in a variety of ways.

We began with a version of the race track problem having a continuous state space. The vehicle could occupy a continuum of places and move at an arbitrary velocity. On a simple example of the racetrack problem (turning a single retangular corner), our DP-based learning scheme using radial basis functions was able to produce successively faster times to the finish line by learning to take the corner at increasingly better trajectories, but learning was very slow. Our research therefore went in two directions: 1) We used a finite-state racetrack problem to compare our DP-based learning algorithms with the conventional solution method (conventional DP). This version of the problem satisfies the conditions required for a convergence theorem we proved. [3]. 2) This problem cries out strongly for the application of a modular architecture in which different modules are switched in for different track configurations. This motivated the study of extending the modular architecture Jacobs [8; 9] to apply to this and similar problems, described below.

4 Modular Architectures

Work on a modular network architecture was begun under the previous AFOSR grant. This work was completed in the period being reported and formed the basis of the Ph.D. dissertation of R. A. Jacobs. This is a method for improving the learning ability of artificial neural networks by organizing several networks into a modular structure [8; 9]. One advantage of such a structure is that the individual networks are not faced with solving large problems in their entirety. Large problems are solved by the combined efforts of several networks. The learning method is a generalization of the unsupervised learning method of competitive learning to the supervised case. After Jacobs was awarded the Ph.D. in May 1990, he worked as a post doctoral researcher at MIT under the direction of Michael Jordan before taking his current position as Assistant Professor of Psychology at the University of Rochester. This work has been very influential in the neural network community, and current work of Jacobs and Jordan continues to develop this basic idea with considerable success.

Whereas Jacobs' architecture is for supervised learning, our own research with modular architectures extended Jacobs' ideas to a modular architecture for reinforcement learning. The ideas was to develop a learning architecture which would facilitate transfer of learning among multiple sequential decision tasks. This is important because sophisticated autonomous agents will have to learn to solve many different tasks, not just one: they should learn throughout their "lives." While achieving transfer of learning across an arbitrary set of tasks is difficult, or even impossible, there are useful and general classes of tasks where such transfer is achievable. We focused on extending DP-based reinforcement learning algorithms to compositionally structured sets of sequential decision tasks. Specifically, we studied learning agents that have to learn to solve a set of sequential decision tasks, where the more complex tasks, called *composite tasks*, are formed by temporally concatenating several simpler, or *elemental*, tasks. Learning occurred under the assumption that a composite task's decomposition into a sequence of elemental tasks was unknown to the learning agent.

Our architecture, called CQ-L, performs *compositional Q-learning*, where Q-learning is a DP-based reinforcement learning method proposed by Watkins [15; 16]. It is a kind of Monte Carlo DP method for estimating the value of performing various actions when the environment is in various states. These values are stored in a function called the Q-function of the task. CQ-L consists of several Q-learning modules, a gating module, and a bias module. In different simulations these modules were variously implemented as lookup tables or as radial basis networks. When trained on a set of compositionally-structured sequential decision tasks, CQ-L is able to do the following: 1) learn the Q-functions of the elemental tasks in separate Q-learning modules; 2) determines the decomposition of the composite tasks in terms of the elemental tasks; 3) learns to construct the Q-functions of the composite tasks by temporally concatenating the Q-functions of the elemental tasks; and 4) learns the constant biases that are added to the Q-value functions of the elemental tasks to construct the Q-value function of the composite tasks.

Simulations using the navigation testbed described above showed that CQ-L is able to learn tasks complex enough to evade solution via a conventional DP-based learning architecture. CQ-L is more powerful than the conventional architecture because it uses solutions of the elemental tasks as building blocks for solving the composite tasks. Transfer of learning is achieved by sharing the elemental task solutions across several composite tasks. This is work of S. P. Singh, a research assistant who has been funded by this grant. Singh has published several papers on his work [14; 12; 13] and is expected to complete the Ph.D. degree in the summer of 1993. Singh's work has already been influential in the AI Machine Learning research community, where increasing attention is being devoted to DP-based reinforcement learning as a component of intelligent agents.

5 Abstract Actions

Closely related to our work with modular architectures is our study DP-based learning with abstract actions. Most applications of DP-based learning described in the literature use these methods at a very low level. For example, the learning component's actions may be primitive movements in a navigation problem. This low level of abstraction generally produces very difficult tasks that can be learned only very slowly. Part of our research effort has been directed toward *raising the level of abstraction at which DP-based learning algorithms are applied.* One way to do this is by letting the learning component's actions be control signals to other system components instead of low-level overt actions in the system's environment. This is one way to incorporate prior knowledge into a learning system in order to improve its performance, and it addresses the problem of having the system perform acceptably while it is learning: If a learning system is to learn from its failures, how can one prevent these failures from producing inconvenient, expensive, or catastrophic results? This issue, perhaps more than any other, has limited the utility of DP-based reinforcement learning in many real-world applications. One answer is to use reinforcement learning as a component of a more complex system.

We experimented with a kind of "bahavior based" reinforcement learning in which the learning component's task is to learn how to coordinate a repertoire of behaviors that have been hand-crafted to 1) achieve desired goals, and 2) avoid catastrophic failure. Learning the right way to compose these behaviors in a state-dependent manner can improve the system's behavior toward optimality while it is operating adequately. We are currently applying these ideas to the navigation domain. The abstract actions correspond to two navigation functions that are computed by using the harmonic function approach to path-planning recently developed by Connolly and Grupen, colleagues doing robotics research at the University of Massachusetts.

In harmonic function path planning, *navigation functions are obtained as solutions of* Laplace's equation (an elliptic partial differential equation) over the relevant robot configuration space. A navigation function is a function with the property that a robot following its gradient from any point in space is guaranteed to reach the goal configuration while avoiding all obstacles. Different boundary conditions of Laplace's equation produce different navigation functions. One such function (obtained using Dirichlet boundary conditions) tends to repel the robot directly away from obstacles while attracting it to the goal. Another navigation (obtained using Neumann boundary conditions) tends make the robot "hug" the obstacle boundaries while attracting it to the goal.

We experimented with using DP-based learning to adjust how these functions were combined to produce another navigation function enabling the robot to reach the goal much faster than it could using either function alone. This can be done in such a way that throughout repeated learning trials, the robot always reaches its goal and never hits an obstacle. Thus learning can occur on-line while the robot is actually performing its designated task without risking inadequate performance. Reinforcement learning is used for perfecting skilled performance, not for achieving adequate performance. We think that reinforcement learning will be most useful in this capacity. We produced successful demonstrations of these ideas in simulated environments, and we are currently applying them to an actual GE P-50 robot arm

6 Theory

We have made considerable progress in increasing our theoretical understanding DP-based reinforcement learning methods and how they relate to other methods. We wrote an extensive paper [3], still under review for *Artificial Intelligence Journal*), that relates these learning algorithms to the theory of asynchronous DP [4] and to the heuristic search method called Learning Real-Time A* [10]. This resulted in a convergence theorem for a class of DP-based algorithms and clearly articulates the advantages they offer over conventional methods for some types of problems. We have also begun development of theory in which some versions of DP-based learning algorithms can be derived as Robbins-Monro types of stochastic approximation methods for solving the Bellman optimality equation. We are currently studying the stochastic approximation literature to derive asymptotic convergence results as well as rate of convergence results.

7 Conclusion

The period covered by this grant has seen a remarkable increase in the number of researchers studying DP-based reinforcement learning. This is due in part to increased interest in the study of embedded autonomous agents. Learning is being widely recognized as an essential capabability of such agents, and DP-based reinforcement learning is directly applicable to the kinds of problems such agents face. Our research funded by this and other grants, as well as the research conducted at other laboratories, is quickly moving these methods toward becoming standard tools that can be successfully applied to a wide range of problems. While the theory of these algorithms is still underdeveloped, we now have a much clearer idea of how they are related to more traditional methods of decision theory and control. We are convinced that DP-based reinforcement learning, in all of its varieties, is a collection of novel algorithms that will find increasing use in forming useful approxmate solutions to stochastic sequential decision problems of practical importance.

References

[1] J. R. Bachrach. A connectionist learning control architecture for navigation. In R. P. Lippmann, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 457–463, San Mateo, CA, 1991. Morgan Kaufmann.

[2] J. R. Bachrach. Connectionist modeling and control of finite state environments. Technical Report 92-6, Department of Computer and Information Science, University of Massachusetts, Amherst, MA, 1992.

[3] A. G. Barto, S. J. Bradtke, and S. P. Singh. Real-time learning and control using asynchronous dynamic programming. Technical Report COINS Technical Report 91-57, University of Massachusetts, Amherst, MA, 1992.

[4] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[5] V. Gullapalli. Modeling cortical area 7a using stochastic real-valued (srv) units. In D. Touretsky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, editors, *Connectionist Models: Proceedings of the 1990 Summer School*, pages 363–368. Morgan Kaufmann, San Mateo, CA, 1990.

[6] V. Gullapalli. Robust control under extreme uncertainty. In *Neural Information Processing Systems 5*, San Mateo, CA, to appear. Morgan Kaufmann.

[7] V. Gullapalli, R. A. Grupen, and A. G. Barto. Learning reactive admittance control. In *1992 IEEE Conference on Robotics and Automation*, Nice, France, 1992

[8] R. A. Jacobs, M. I. Jordan, and A. G. Barto. Task decomposition through competition in a modular connectionist architecture: The what and where vision task. *Cognitive Science*, 15:219–250, 1991.

[9] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

[10] R. E. Korf. Real-time heuristic search. *Artificial Intelligence*, 42:189–211, 1990.

[11] S.P. Singh. The efficient learning of multiple task sequences. In J.E. Moody, S.J. Hanson, and R.P. Lippman, editors, *Advances in Neural Information Processing Systems 4*, pages 251–258, San Mateo, CA, 1992. Morgan Kaufmann.

[12] S.P. Singh. Reinforcement learning with a hierarchy of abstract models. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 202–207, Menlo Park, CA, 1992. AAAI Press/MIT Press.

[13] S.P. Singh. Scaling reinforcement learning algorithms by learning variable temporal resolution models. In *Proceedings of the Ninth International Machine Learning Conference*, pages 406–415, San Mateo, CA, 1992. Morgan Kaufmann.

[14] S.P. Singh. Transfer of learning by composing solutions for elemental sequential tasks. *Machine Learning*, 8:323–339, 1992.

[15] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, Cambridge, England, 1989.

[16] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

PUBLICATIONS: 1990–PRESENT

Publications in reviewed journals

M. C. Mozer and J. R. Bachrach. Discovering the structure of a reactive environment by exploration. *Neural Computation*. **2:** 447-457, 1990.

V. Gullapalli. A stochastic reinforcement learning algorithm for learning real-valued functions. *Neural Networks*. **3:** 671-692. 1990.

R. A. Jacobs, M. I. Jordan and A. G. Barto. Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. *Cognitive Science*. **15:** 219-250. 1991.

M. C. Mozer and J. Bachrach. SLUG: A connectionist architecture for inferring the structure of finite-state environments. *Machine Learning* (Special Issue on Connectionist Approaches to Language Learning) **7**(2 3): 139-160, 1991.

S.P. Singh. Transfer of learning by composing solutions for elemental sequential tasks. *Machine Learning*. **8:** 323-339. May 1992.

Refereed conference proceedings

A. G. Barto, R. S. Sutton and C. J. C. H. Watkins. Sequential decision problems and neural networks. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*. pp. 686-693. Morgan Kaufmann Publishers, San Mateo, CA, 1990.

M. C. Mozer and J. Bachrach. Discovering the structure of a reactive environment by exploration. In D. S. Touretzky, editor. *Advances in Neural Information Processing Systems 2*. pp. 439-446. Morgan Kaufmann: San Mateo, CA, 1990.

R. C. Yee, S. Saxena, P. E. Utgoff and A. G. Barto. Explaining temporal differences to create useful concepts for evaluating states. In *Proceedings of the 8th National Conference on Artificial Intelligence*. pp. 882-888. AAAI Press MIT Press, 1990.

A. G. Barto, R. S. Sutton and C. J. C. H. Watkins. Sequential decision problems and neural networks. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pp. 686-693. Morgan Kaufmann Publishers, San Mateo, CA, 1990.

M. I. Jordan and R. A. Jacobs. Learning to control an unstable system with forward modeling. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pp. 324-331. Morgan Kaufmann Publishers, San Mateo, CA, 1990.

M. C. Mozer and J. Bachrach. Discovering the structure of a reactive environment by exploration. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*. pp. 439-446. Morgan Kaufmann Publishers, San Mateo, CA, 1990.

J.R. Bachrach. A connectionist learning control architecture for navigation. In R. Lippmann, J. Moody and D. Touretzky, editors, *Advances in Neural Information Processing 3*. Morgan Kaufmann: San Mateo, CA, 1991. pp. 457-463.

S.P. Singh. Transfer of learning across compositions of sequential tasks. In L.A. Barnbeum and G.C. Collins, editors, *Machine Learning: Proceedings of the Eighth International Workshop* (ML91), Morgan Kaufmann: San Mateo, CA, 1991. pp. 348-352.

V. Gullapalli. A comparison of supervised and reinforcement learning methods on a reinforcement learning task. *Proceedings of the 1991 IEEE International Symposium on Intelligent Control,* Arlington, VA, August 1991.

V. Gullapalli. Associative reinforcement learning of real-valued functions. *Proceedings of the 1991 IEEE International Conference on Systems, Man, and Cybernetics,* Charlottesville, VA, October 1991.

V. Gullapalli, R. A. Grupen and A. G. Barto. Learning reactive admittance control. In *Proceedings of the 1992 IEEE Conference on Robotics and Automation,* Nice, France, May 1992.

S. P. Singh. Scaling reinforcement learning algorithms by learning variable temporal resolution models. In *Proceedings of the Ninth Machine Learning Conference,* Aberdeen, Scotland, 1992. Morgan Kaufmann, pp. 406-415.

V. Gullapalli. Associative reinforcement learning of real-valued functions. *Proceedings of the 1991 IEEE International Conference on Systems, Man, and Cybernetics,* Charlottesville, VA, October 1991.

S. P. Singh. Reinforcement learning with a hierarchy of abstract models. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92),* San Jose, CA, July 1992. AAAI Press/MIT Press, pp. 202-207.

S. P. Singh. The efficient learning of multiple sequential tasks. In J.E. Moody, S.J. Hanson and R.P. Lippman, editors, *Advances in Neural Information Processing 4.* Morgan Kaufmann: San Mateo, CA, 1992, pp. 251-258.

A. G. Barto and S. J. Bradtke. Learning to solve stochastic optimal path problems using real-time dynamic programming. *Proceedings of the Seventh Yale Workshop on Adaptive and Learning Systems,* New Haven, CT, May 1992.

V. Gullapalli. Robust control under extreme uncertainty. In *Neural Information Processing Systems 5,* Morgan Kaufmann: San Mateo, CA, to appear.

S. J. Bradtke. Reinforcement learning applied to linear quadratic regulation. In *Neural Information Processing Systems 5,* Morgan Kaufmann: San Mateo, CA, to appear.

Book chapters published

A. G. Barto and S. P. Singh. Reinforcement learning and dynamic programming. In *Proceedings of the Sixth Yale Workshop on Adaptive and Learning Systems,* held August 15-17, 1990 in New Haven, CT.

A. G. Barto and S. P. Singh. On the computational economics of reinforcement learning. In D.S. Touretzky, J.L. Elman, T.J. Sejnowski and G.E. Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School.* San Mateo, CA: Morgan Kaufmann, 1990. pp. 35-44.

V. Gullapalli. Modeling cortical area 7a using stochastic real-valued (SRV) units. In D.S. Touretzky, J.L. Elman, T.J. Sejnowski and G.E. Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann, 1990.

R. S. Sutton and A. G. Barto. Time-derivative models of Pavlovian reinforcement. In *Learning and Computational Neuroscience*, M. Gabriel and J. Moore, editors, The MIT Press, Cambridge, MA, 1990, pp. 497-537.

A.G. Barto. Some learning tasks from a control perspective. In L. Nadel and D. Stein, editors, *1990 Lectures in Complex Systems*. Addison-Wesley, 1991, pp. 195-223.

R. S. Sutton, A. G. Barto and R. J. Williams. Reinforcement learning is direct adaptive optimal control. Proceedings of the 1991 American Control Conference, June 26-28, Boston, MA, pp. 2143-2146.

V. Gullapalli. Dynamic systems control via associative reinforcement learning. In B. Soucék, editor, *Dynamic, Genetic, and Chaotic Programming: The Sixth Generation*. New York, NY: John Wiley & Sons, 1992.

A. G. Barto. Reinforcement learning and adaptive critic methods. In *Handbook of Intelligent Control*, D.A. White and D.A. Sofge, editors. New York: Van Nostrand Reinhold, 1992, pp. 469-491.

Technical reports

A. G. Barto, S. J. Bradtke and S. P. Singh. Real-time learning and control using asynchronous dynamic programming. Technical Report 91-57, Computer Science Dept., University of Massachusetts, Amherst, August 1991. (Submitted to *Artificial Intelligence Journal*.)

A. G. Barto and V. Gullapalli. Neural networks and adaptive control. NPB Technical Report 6, Center for Neuroscience Research on Neuronal Populations and Behavior, Northwestern University, March 1992. [To appear in P. Rudomin, M.A. Arbib and F. Cervantes-Perez, editors, *Natural and Artificial Intelligence*, Research Notes in Neural Computation, Springer-Verlag (in press).]

R. Yee. Abstraction in control learning. COINS Technical Report 92-16, University of Massachusetts, March 1992.

A. G. Barto, S. J. Bradtke and S. P. Singh. Learning to act using real-time dynamic programming. CMPSCI Technical Report 93-02, University of Massachusetts, January 1993. (Supercedes TR 91-57.) Submitted to *AI Journal*.

Graduate students

Jonathan Bachrach
Robert Crites
Vijaykumar Gullapalli
Robert Jacobs
Satinder Singh
Richard Yee

Theses produced:

R. A. Jacobs. Task Decomposition Through Competition in a Modular Connectionist Architecture. (Ph.D. Thesis) COINS Technical Report 90-44, University of Massachusetts at Amherst, May 1990.

J. R. Bachrach. Connectionist Modeling and Control of Finite State Environments. (Ph.D. Thesis) COINS Technical Report 92-6, University of Massachusetts, Amherst. January 1992.

V. Gullapalli, Reinforcement Learning and its Application to Control. (Ph.D. Thesis) COINS Technical Report 92-10, University of Massachusetts, Amherst. January 1992.

External honors, etc.

Andrew G. Barto became a Senior Fellow of IEEE.

Andrew G. Barto gave an invited plenary address entitled "Learning to Act: A Perspective from Control Theory" at the Tenth Annual Meeting of the American Association for Artificial Intelligence (AAAI-92) at San Jose, CA, July 15, 1992.

Andrew G. Barto gave the invited plenary lecture, entitled "Reinforcement Learning," at the 1992 Conference on Learning Theory at the University of Pittsburgh, July 27, 1992.